DRAWING FIGURES

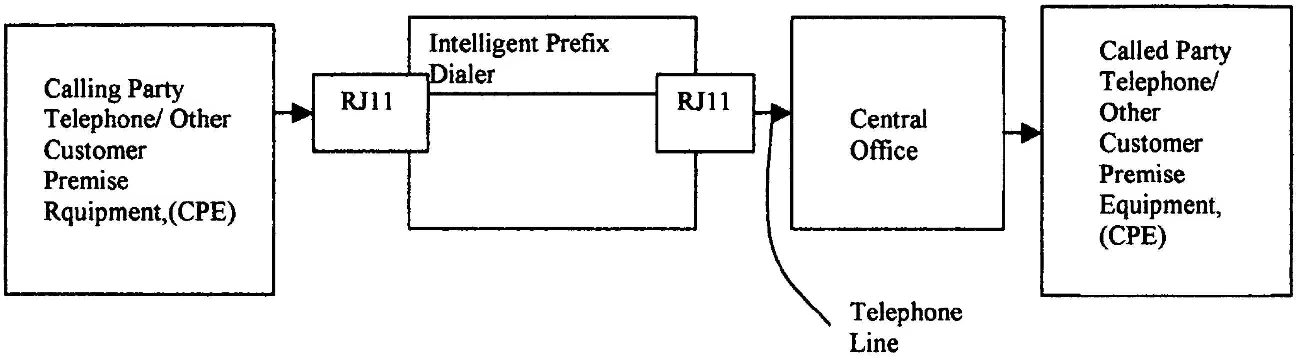

Calling Party Telephone/ Other Customer Premise Rquipment,(CPE)
RJ11
Intelligent Prefix Dialer
RJ11
Central Office
Telephone Line
Called Party Telephone/ Other Customer Premise Equipment, (CPE)

FIGURE 1

Line Interruption Circuit Detailed Configurateion

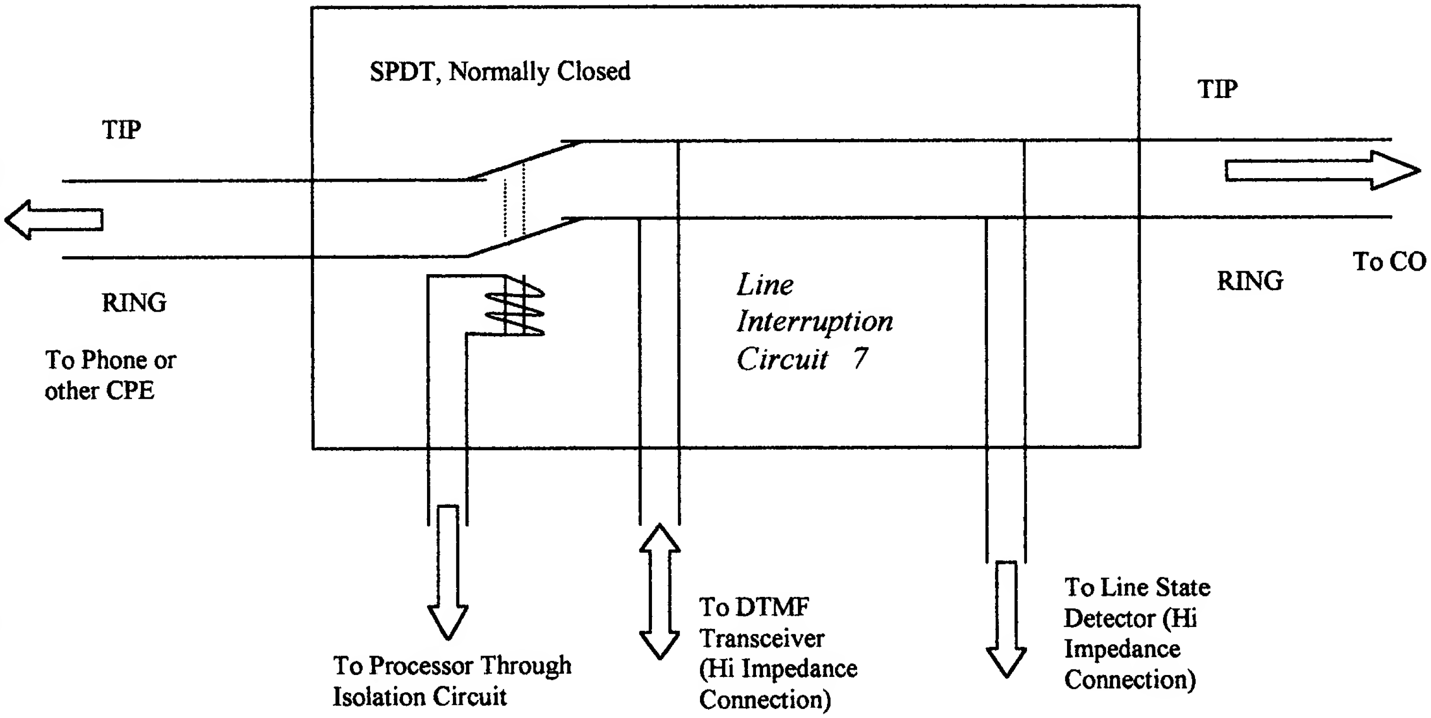

SPDT, Normally Closed
TIP
RING
To Phone or other CPE
TIP
RING
To CO
Line Interruption Circuit 7
To Processor Through Isolation Circuit
To DTMF Transceiver (Hi Impedance Connection)
To Line State Detector (Hi Impedance Connection)

Figure 2a

Intelligent Telephone Prefix Dialer, standalone POTS environment

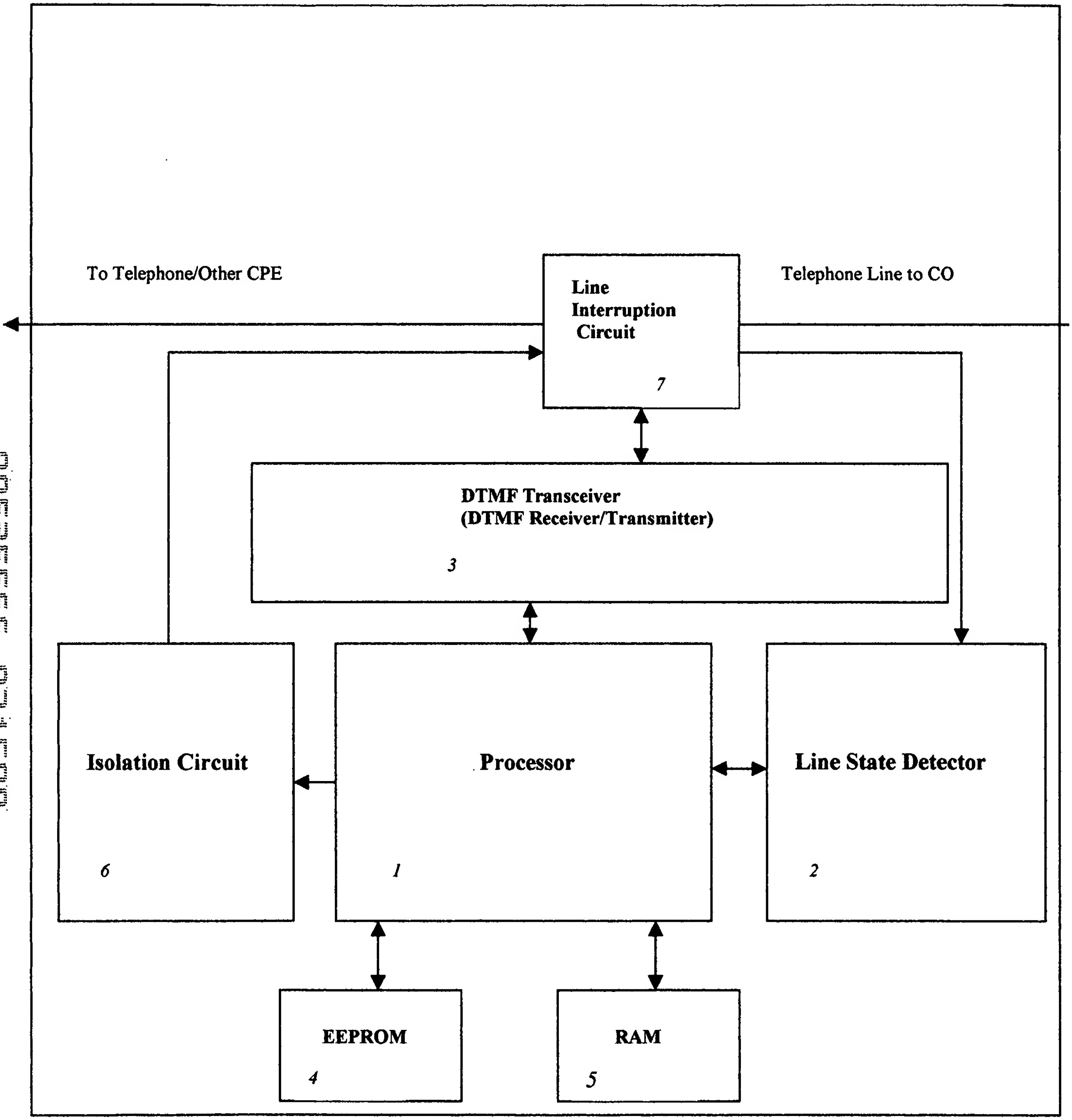


Figure 2b

Intelligent Telephone Prefix Dialer embedded in a POTS Telephone Set

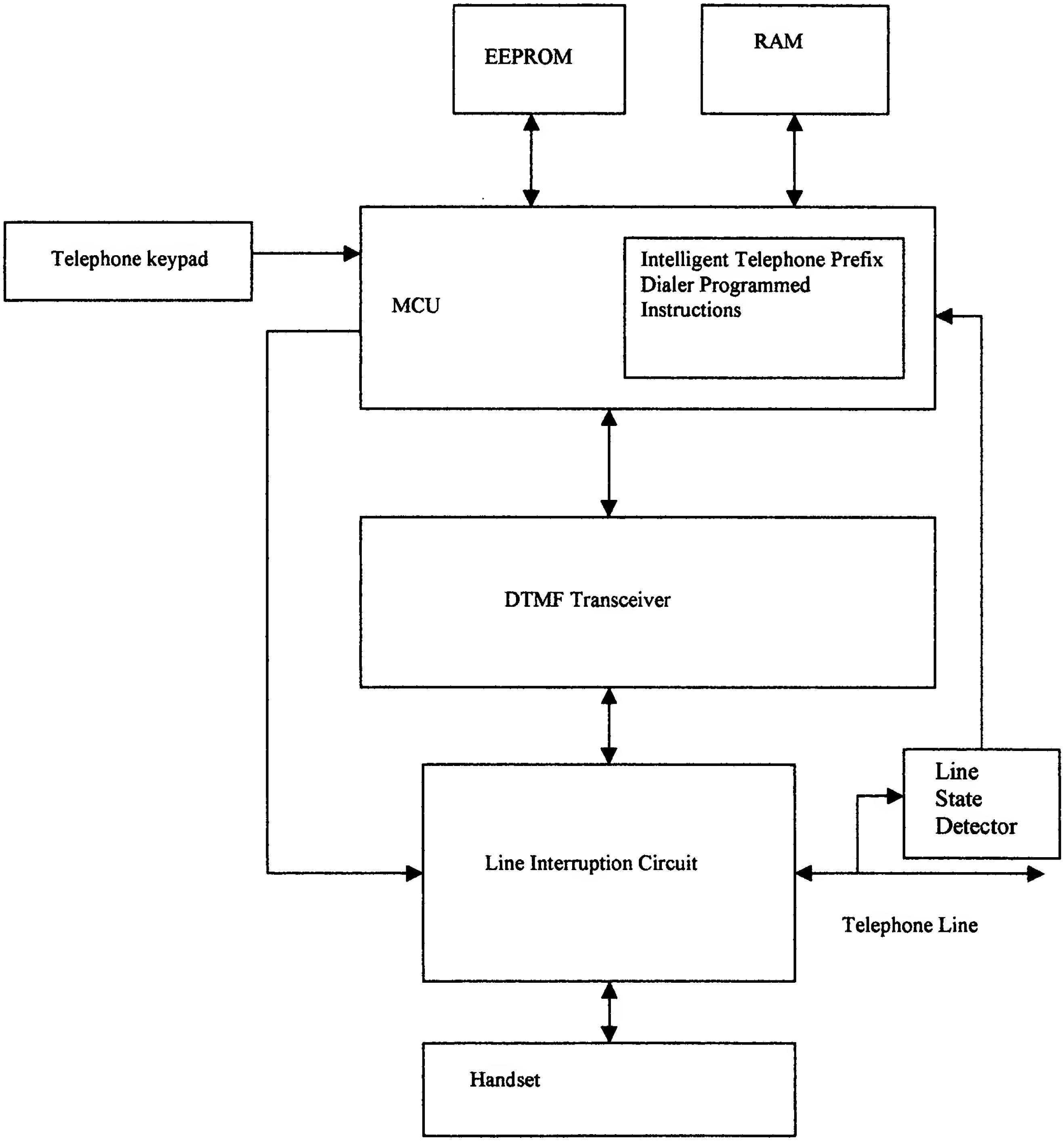

EEPROM
RAM
Telephone keypad
MCU
Intelligent Telephone Prefix Dialer Programmed Instructions
DTMF Transceiver
Line Interruption Circuit
Line State Detector
Telephone Line
Handset

Figure 3

Intelligent Telephone Prefix Dialer embedded in an ISDN telephone set

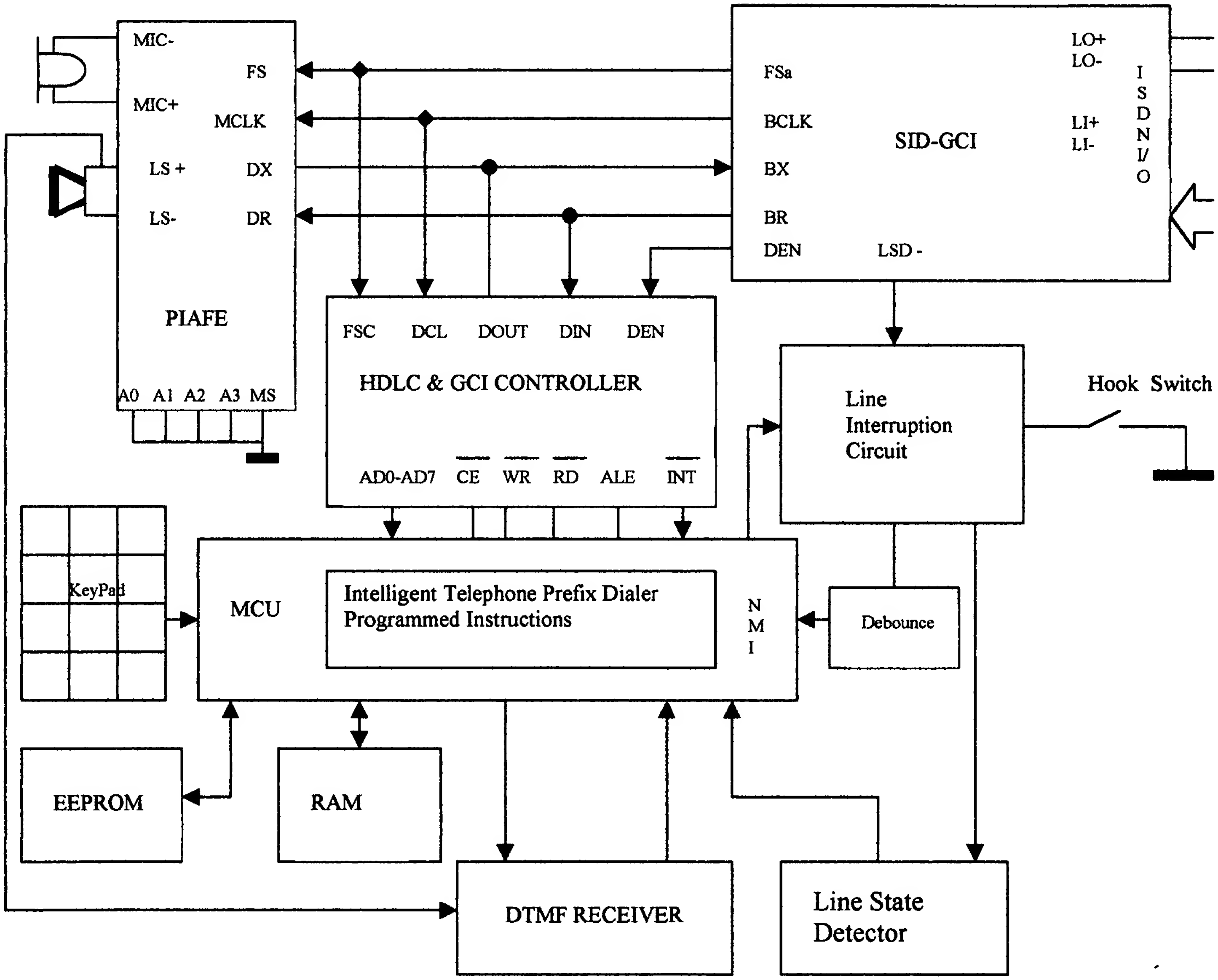
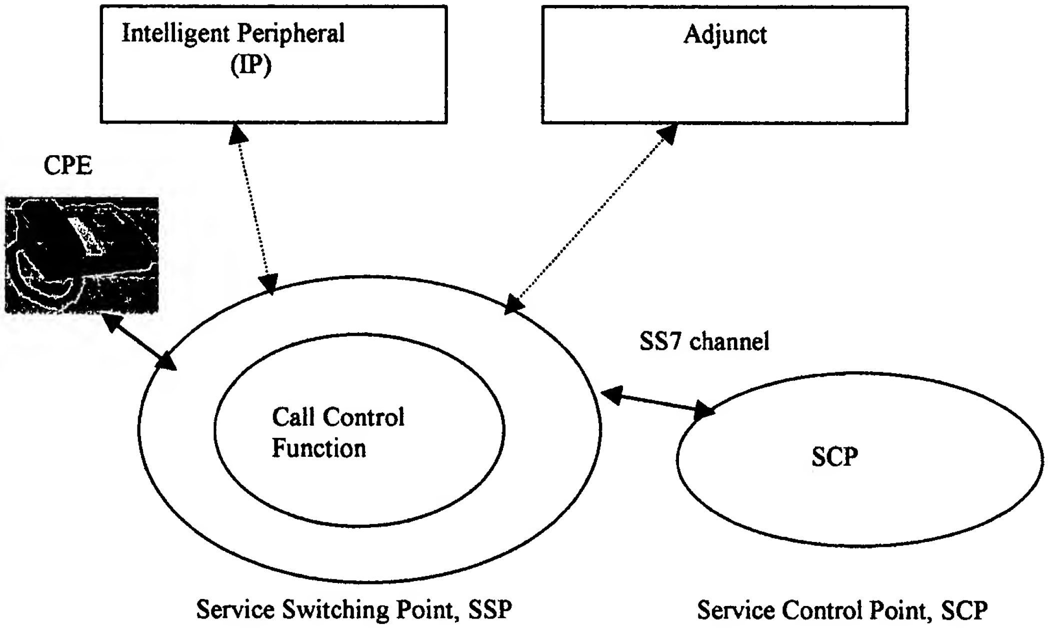


Figure 4

Intelligent Prefix Dialer Integrated into Service Provider's Advanced Intelligent Network Equipment


Intelligent Peripheral (IP)
Adjunct
CPE
SS7 channel
Call Control Function
SCP
Service Switching Point, SSP
Service Control Point, SCP

Control Logic at SCP

- Analyze data triggered from Call Control at SSP
- If redial with prefix is required, append user dialed number to default dial prefix in user options database and send resultant digit sequence back to SSP
- else send "dial as is" message to SSP

Call Control at SSP, IP:

- Capture and count dtmf digits
- Time interdigit delay
- Timeout to close digit sequence
- Report results to SCP
- Receive Dial String from SCP
- Dial Digits to complete call

User Options Interface at SSP/IP

- Capture star code for Intelligent Prefix Dialer options setup and send to SCP
- Receive dial tone stutter prompt from IP, capture default prefix from user, and send codes to SCP
- Receive dial tone stutter/voice acknowledgement from IP

User Options Recording at SCP

- Store default prefix in user options database for the Intelligent Prefix Dialer Service
- Send Acknowledge to SSP

Figure 5

INTELLIGENT TELEPHONE PREFIX DIALER PSEUDOCODE

Version Beta 3.0

```
Subroutines
          DISPLAYPREFIX@
  LINEMONITOR@
          MONITORLINE@
          CAPTUREDIGITS@
          CAPOPTIONSTRINGS@
          CAPREFIXSTRING@
          FLASHLINE@
          GETNDX@
          CHECK_FOR_TEN@
          DIALNUMBER@
          PARSEOPTIONS@
          PARSESTRING@
Data
          LENGTH                           /* length of table*/
          TABLE                            /*start of table*/
          SUM                              /*sum of digits*/
          COUNT                            /*count of digits*/
          TELNO(8)                         /*user dialed digits*/
          PREFIX                           /*user defined dial prefix*/
          DIALTONE_FLAG                    /*Flag to indicate line state */
                                           /* On Hook = 0, Off Hook = 1*/
                                           /* Line one to Off Hook Line two*/
          DIAL_STRING(10)                  /*The reparsed dial string necessary to complete */
                                           /* the call*/
          USER_REQUEST_FLAG                /*Flag to initiate user input of prefix code*/
          NDX                              /*# Pointer for user TELNO entries
                                           /*intoDIALSTRING*/
          NUMBER_OF_DIGITS_CAPTURED        /*number of digits received by dtmf receiver before*/
                                           /*timeout*/
          ON_HOOK_TIME_COUNTER             /*amount of time that receiver is on hook*/
          BYPASS                           /*bypass bit, if set to 1, bypasses flashook 2 and 3*/

/***************************************************************************/
```

```
Program MAIN
                                                          /*Declare and initialize all variables*/
Declare and Intitilize Hardware specific variables for dtmf transceiver and other hardware
Dtmf                                  var     byte
Bypass                                var     byte
Dt_flag                               var     bit
Dt_det                                var     INL.bit2          /*Detect bit from dtmf receiver*/
Dialtone_flag     var     bit
Number_of_Digits_Captured             var     byte              /*Range index to telno()*/
Digit             var     byte                /*Index  of digits to dial by autodialer*/
I                 var     word
L                 var     byte
K                 var     bit
Ndx               var     nib


Gosub  GETNDX                   /*Get ndx from EEPROM*/
For I = 1 to ndx - 1
Get prefix code from EEPROM and place into dial_string(I)
next
GOSUB DISPLAYPREFIX   /*Show the stored dialing prefix*/
CAPDIGITS:
        GOSUB CAPTUREDIGITS  /*Start listening for dial string digits entered by user*/
              If NUMBER_OF_DIGITS_CAPTURED <> (10 – NDX)  + 1 then
                goto INHIBITDIAL
              fi


   GOSUB PARSESTRING                 /*Parse the TELNO() into DIAL_STRING()
             Pause 160                     /*Time delay before initiating flash hook sequence*/
       GOSUB FLASHLINE               /*First Flash hook*/
             Pause 700                     /*Time delay before further action*/
             If BYPASS =1 then GOTO SKIP_FLASHES /*2nd and 3rd flash only necessary for 3
                                                          /*way call*/
   GOSUB FLASHLINE                   /* 2nd Flash hook*/
             Pause 700                     /*Time delay before further action*/
   GOSUB FLASHLINE                   /* 3rd  Flash hook*/
             Pause 700                     /*Time delay before further action*/
SKIP_FLASHES:
```

```
            pause 700                /*Time delay before initiate redial*/
      GOSUB DIALNUMBER               /*Dial the number with the required prefix*/
INHIBITDIAL:
          GOSUB LINEMONITOR                  /*Stay put until line goes onhook*/
      GOSUB MONITORLINE              /*Stay put until line goes offhook*/

      GOTO CAPDIGITS         /*Start listening for digits again*/



/**************************************************************************/

SUBROUTINE:LINEMONITOR
LOOPDT1:
        Set DIALTONE_FLAG from (Telephone Line) /*0 is ONHOOK, 1 is OFFHOOK*/
                IF DIALTONE_FLAG indicates OFFHOOK then  GOTO LOOPDT1
        Return

/**************************************************************************/




/**************************************************************************/
SUBROUTINE:MONITORLINE
        Initialize ON_HOOK_TIME_COUNTER to Zero
LOOPDT2:
        Set DIALTONE_FLAG from (Telephone Line) /*0 is ONHOOK, 1 is OFFHOOK*/
                IF DIALTONE_FLAG indicates ONHOOK then
                  Do
                    Increment ON_HOOK_TIME_COUNTER
                    GOTO LOOPDT2
                  Done
                fi
                IF ON_HOOK_TIME_COUNTER > 800 then set BYPASS to 1
                fi
        Return
```

```
/**************************************************************************/

/**************************************************************************/
SUBROUTINE: CAPTUREDIGITS
CAPTUREDIGITS:
        SETUP dtmf hardware for dtmf  READ
        For I = 1 to 1700    /*Initialize Interdigit count down timer*/
         Get DIALTONE_FLAG from (Telephone Line) /*If not still OFFHOOK then EXIT to MAIN*/
          If DIALTONE_FLAG = 0 then GOTO MAIN
                fi
          POLL for dtmf tone from (DTMF RECEIVE CHIP)
          If tone not detected then NEXT I          /*Increment Interdigit count down timer*/
          else
           Increment NUMBER_OF_DIGITS_CAPTURED
           If NUMBER_OF_DIGITS_CAPTURED > (10 – NDX)  + 1 then GOTO MAIN
                              /*user dialed more than */
                              /*prefix digits plus user digits and does not need help here */
           READ dtmf tone into variable DTMF
         TELNO(NUMBER_OF_DIGITS_CAPTURED) = DTMF
        NEXT I
                /*Interdigit Timer has timed out, Check for number of digits received*/
        IF NUMBER_OF_DIGITS_CAPTURED  <  (10 – NDX) + 1  then
          Do
             If  telno(1) = 12 and telno(2) = 1 then
               Do                              /*User has requested to input options*/
                  Gosub PARSEOPTIONS
                  Goto MAIN                    /*Initialize with new user options*/
               Done
            Set NUMBER_OF_DIGITS_CAPTURED =  0
          Done
        Return
/**************************************************************************/
SUBROUTINE: PARSESTRING
          For j = NDX to 10
          DIAL_STRING(j) = TELNO(j – (NDX – 1))
          Next j
```

```
 Return
/**************************************************************************/
/**************************************************************************/
SUBROUTINE: FLASHLINE
                Go ONHOOK
                Pause 600 msec                  '600 milliseconds, nominal, can be between 400 and
                                        '700ms
                Go OFFHOOK
        Return
/**************************************************************************/

/**************************************************************************/
SUBROUTINE: DIALNUMBER
        IF PRIVACY_BIT = 1 then
         Do
           DTMFOUT(*67)              /*Dial the Caller ID Block Code */
         Done
        IF PRIVACY_BIT = 0 then
          Do
           DTMFOUT(*82)              /*Dial the Caller ID Send Code*/
          Done
        IF ONE_PLUS_BIT = 1 then
          Do
           DTMFOUT(1)                /*Dial 1 before the area code, etc*/
          Done
        For DIGIT = 1 to 10
        DTMFOUT(DIALSTRING(DIGIT)) /*Dial the prefix code and the rest of the
                                               /*phone number*/
        Return
/**************************************************************************/

/**************************************************************************/

SUBROUTINE: PARSEOPTIONS
        Write to DisplayDevice("PRIVACY?: Y/N) /*Prompt for user to turn Call ID Block ON or */
                                          /*OFF*/
        Gosub CAPOPTIONSTRINGS      /*Get user input*/
        Write user input to EEPROM
         Read user input from EEPROM
         Write user input from EEPROM to DisplayDevice /*User selection confirmed on */
                                                          /*DisplayDevice*/

        Write to DisplayDevice("1 PLUS ON?: Y/N) /*Prompt for user to turn 1 PLUS Dialing
                                                           /*ON or OFF*/
        Gosub CAPOPTIONSTRINGS      /*Get user input*/
        Write user input to EEPROM
         Read user input from EEPROM
         Write user input from EEPROM to DisplayDevice /*User selection confirmed  on*/
                                                          /*DisplayDevice*/

        Write to DisplayDevice("ENTER PREFIX# ) /*Prompt for user to enter dialing prefix*/
        Gosub CAPREFIXSTRING         /*Get user input of dialing prefix*/
        Write user input to EEPROM
        While user input from EEPROM <> 12
          Do
            Read user input from EEPROM
            Gosub CHECK_FOR_TEN
            Write user input from EEPROM to DisplayDevice /*User entry confirmed  on*/
                                                          /*DisplayDevice*/
    Done
        Return
/***************************************************************************/


/***************************************************************************/

SUBROUTINE: DISPLAYPREFIX
        READ PrefixData from EEPROM
        WRITE PrefixData from EEPROM to DisplayDevice
Return
/***************************************************************************/

/***************************************************************************/
SUBROUTINE: CAPOPTIONSTRINGS
```

```
        For I =1 to 1900            /* Time out if no user input*/
          When data present from DTMFreceiver
      Do
                READ data from DTMFreceiver into option_bit
                Return
            Done
        Next
Return
/***************************************************************************/

/***************************************************************************/
SUBROUTINE: CAPREFIXSTRING
        Mu = 0
        For I =1 to 1900            /* Time out if no user input*/
          When data present from DTMFreceiver
         Do
                Mu = mu + 1
                READ data from DTMFreceiver into telno(mu)
                If  telno(mu) = 12  or  mu > 7 then
                  Return
                fi
            done
        Next
        Return
/***************************************************************************/
```

```
/***************************************************************************/
SUBROUTINE: GETNDX
for i = 1 to 7
 read from start of  prefix data from EEPROM into digit
 if digit = 12 then ret_ndx
next
return
```

```
ret_ndx:
ndx = I
return
/****************************************************************************/
/***************************************************************************/
SUBROUTINE: CHECK_FOR_TEN
if telno(i) = 10 then zeroit
return
zeroit:
telno(i) = 0                              /*Format output for DisplayDevice*/
return
/***************************************************************************/
```

Programmer Application Notes:

1. Actual programming language used was Parallax, Inc. PBASIC
2. Processor used was the Parallax, Inc. (www.parallaxinc.com), BASIC Stamp II, BS2-IC
3. The Pause instruction argument is in milliseconds
4. The processor clock speed is approximately 20MHZ
5. The PBASIC interpreter executes approximately 3000 instructions per second, i.e. 0.3 milliseconds per instruction. Use the 0.3 milliseconds/instruction value to calculate timeouts and delays that are implemented using loops.
6. Contact the inventor at www.dimensional.com/~jbreki/dialer.html for future development and application notes.

Figure 6